**Paper No.T2.4-15**
05836

# VERIFICATION OF AN ONLINE TEST BY TRACKING EYE MOVEMENT OF THE EXAMINEE

**Nenad Gligorov[1], Aleksandar Kupusinac[2], Dragan Ivetić[3]**
*[1,2,3]Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, Srbija*
[1]ngligorov@uns.ac.rs, [2]sasak@uns.ac.rs, [3]ivetic@uns.ac.rs

***Abstract:*** *With the introduction of online testing arose the issue of high importance, which is making sure that the examinee has understood the question properly. It can be represented by the time the examinee has spent looking at it in order to understand it. Based on the information gathered, the examiner has better insight into the success of the examinee's understanding of the test. This information is crucial for the examiner because it allows them to improve the test in the next iteration: make the question text more understandable, add possible missing answers, more closely describe what answer is expected from an examinee, etc. A potential solution to this issue would be creating an online testing platform that would allow examinees to answer questions while their eye movement is being tracked by an eye tracker camera. The examiner would have the possibility to define regions of interest that represent parts of the questions that need verification, f/e question text, or available answers. This paper analyzes the process of collecting information based on how well and correctly formulated the questions are – based on how much time the student has spent reading the set of questions and/or offered answers. In order to make the whole platform semantic web compliant, regions of interest will be represented by an OWL ontology.*

***Key Words: Online Testing / Eye Tracker / Regions of Interest / Semantic Web / OWL Ontologies***

## 1. INTRODUCTION

Since COVID-19 regulations took place all over the world, most of the exams and tests that were previously held in-person are now being performed online. With that in mind, there is an issue if the examinee is understanding the test correctly. While examinees are able to ask for clarification from the examiner during the in-person exam, that might not always be the case with online exams.

The issue at hand is determining whether the exam was created at sufficient comprehension. This concludes determining the conciseness and the concretization of the questions in the test, how are they placed on the screen, whether the text of the question is clear or not, etc. The time used by the examinee to read and understand the question could come in handy here. It can be measured by tracking eye movement of the examinee, while they are reading a question or its part.

Eye tracking could be performed using the eye tracker – a special camera placed on the examinees computer which tracks their eye movement. Based on the examinee's gaze, valuable information about understanding, solving, inconsistencies and other difficulties about the exam can be obtained. Of course, the answer to a question is equally important. If the question is poorly worded, the answer itself cannot be expected to be sufficient. The spatial layout of the question plays a big role. For example, a question could have a picture or a diagram alongside to the text. If said figure is far away from the text or placed poorly in some other way, the examinee would be forced to switch their gaze rapidly across the screen multiple times, thus reducing the ability to focus on the question and simultaneously increasing the time needed to understand it properly.

All this information is of great value, since it can give insight on how the test should be refined in the future, making every iteration of the test better than the one before. To make all this possible, an online platform for test creation and verification is created. The examiner will be given the ability to create an exam, choose between open and closed question. Most importantly, they will be able to define regions of interest on every question. These regions are meant to be parts of the question that need to be monitored.

After every test, this platform will produce information about the percentage of the test solved correctly, as well as information about the periods of time used for reading every question or some part of it. This information can later help the next generation of examinees understand and solve the test with better sufficiency.

Next chapter of this paper will reflect on the related work, how it was done, what was their result, how it compares to this platform. The central place of the paper will address the implementation and the main idea behind using regions of interest.

## 2. RELATED WORK

There are several scientific papers that correlated to eye tracking while performing an exam or a test. Those papers gave a crucial insight on aspects that are part of the problem that we are solving in this paper. Therefore, they will be mentioned in this chapter.

Eye movement tracking is also used in learning courses, authors of [1] discussed if eye movement tracking as a tool is justified. They analyze multiple eye tracking based measures and their correlation to cognitive comprehension of students who are learning to program. Precisely, intrinsic cognitive load (ICL) is being measured by tracking students' gaze. Authors were then trying to find a connection between these measures with program comprehension. For this experiment, thirty one students were tested on two program codes written in C++. Their task was to find logical (LER) and syntax (SER) errors, these two being two different cognitive tasks. It is important to notice that the students had to perform these tasks without the

help of an Integrated Development Environment (IDE), so they would have to concentrate solely on the code, without the intrusion of tracking processes within the IDE.

Authors mention different eye-based measures such as changes in pupil size, fixation duration, blink rate, and saccade. Within these measures, they focus on fixation duration (FDA) and saccade amplitude average (SAA). Saccade amplitude average is being described as "the sum of all saccade amplitudes divided by the number of saccades in the trial". The result of this experiment was that FDA and SAA greatly differed, depending on which of the two tasks is being performed. Also, the authors found that higher ICL is being associated with longer FDA and shorter SAA. Given that this paper did not actually discuss the implementation of a platform simial to ours, it gave information about eye movement tracking needed to proceed with our paper.

In their study [2], authors use eye tracking to examine the way learners react to instructional videos if there are cues added. The authors orchestrated an experiment with one hundred and twenty college students. Those students were required to watch short instructional videos without cues and also when there are cues. Cues themselves are separated into three categories: visual cues, textual cues, combined textual and visual cues.

This study focuses on showing "how eye-tracking technology can be used to provide evidence and efficacy of the additional pointing or highlighting cues". The authors focus on learners' gaze fixation to determine added cues efficiency. After the video, learners are given a questionnaire to determine the amount of knowledge obtained. Authors conclude the paper by saying that learners memorized most efficiently with the combination of textual and visual cues. While this study also did not mention what kind of an eye tracker camera was used, it gave insight on the importance of using eye movement tracking to refine instructional videos which can be transferred to online tests as well.

Authors of [3] conducted an analysis on an online networks exam while using eye tracking techniques on students. They used Eye Tribe tracer, as opposed to Gazepoint Eye Tracked being used in this paper. Different groups of students were tested, while the groups were created depending on student's prior knowledge of the subject. In this study, as well as of the above, the authors focused on the duration of fixations and saccades lengths.

## 3. ABOUT THE PLATFORM

The online platform developed is a web application that enables the creation of new tests that are later distributed to examinees for solving. After the test is finished, the percentage of efficiency of the test is displayed on the screen. Students can see correctly answered questions, as well as the questions where they answered wrongly with the mark on the correct answer.

For the user to be able to access the application, they need to log in first. Since this platform is designed as a tool for schools and universities, there is no registration functionality, with the reason for that being security. Situations in which a third party, that is, those persons who do not belong to an educational institution, register on the system and access tests, whether it is creating them or solving them, is a huge attack on the platform security and the legitimacy of created tests.

Instead of using user log-in, the main idea was that the student accounts be written directly into the database by a system administrator, which is the usual way of working for this type of application.

Platform users can have one of the two following roles: professor, student.

If the user is logged in as a professor, the application enables them to create tests, as well as keep track of the results, once the exam is finished.

Once the test creation functionality has been chosen, the user is required to enter a name for the test. This name has to be unique, that is, there must not be another test with the same name. After choosing the name which has been proven to be unique, the professor can create questions within the test. Two options are given: essay questions and multiple-answers questions. With the essay questions, the professor is required to enter text of the question, while students later have to answer it – in the form of an essay. With the multiple essay questions, the professor has to define the question text and four possible answers. Here, they also have to mark one or more possible answers to be correct. Creating a question when none of the possible answers are correct is not possible. On the other side, it is possible to define a question where every possible answer is correct.

Figure 1 shows the interface used to create a question with multiple possible answers. As it is almost useless to only track X and Y coordinates while the student is solving the test, the professor can define regions of interest (ROI). In this way, the user can decide which parts of the test are important for future analysis.

With a click of the mouse and moving the cursor, a transparent rectangle with red edges is created. This rectangle is representing the borders of the newly created ROI. By releasing mouse click a new ROI is created and a form is shown to the user. In this form, they are required to enter a name for the ROI. This name is later shown in the top left corner of the rectangle. If a mistake has been made while defining the size of a new ROI, the user can cancel the action by pressing the letter "P" on the keyboard.

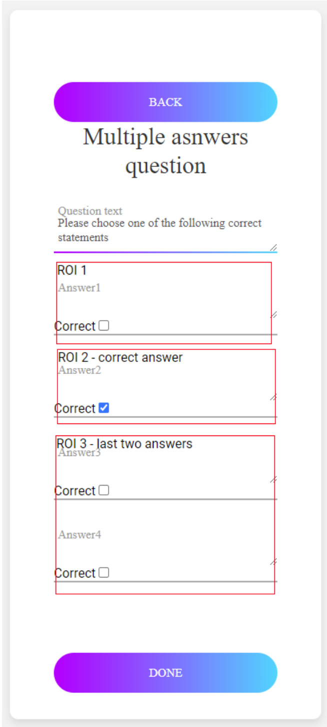


*Figure 1. User interface for creating a multiple-answers question*

Three regions of interest are shown in Figure 1. The first one – with the name “ROI 1” covers the region of question 1. The second one – with the name “ROI 2 – correct answer” covers the region of the second question. The third, and last, ROI with the name “ROI 3 – last two answers” covers the region of the last two answers. It is important to notice that ROIs are never shown to the students while the test is in progress. Even though they are not graphicly shown, the application still tracks student’s gaze and notes when it enters one of the defined regions of interest.

A user with the role of professor has one last functionality available, which is downloading a report after the test is finished. By choosing this option, the user is presented with a screen that shows a heatmap with all the ROIs in the test, as well as the time every student has spent fixating on it. This screen is shown in Figure 2.

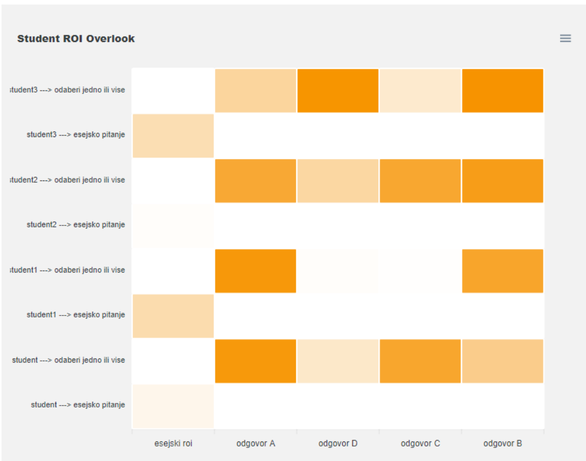


Figure 2. Heatmap of the students' fixation while solving the test

By hovering the cursor over a field, information about the coordinates and the time spent fixating by the student is shown. It is also possible to download this information in one of the following formats: SVG, PNG and CSV.

If a user is logged in as a student, the application will give them the ability to choose one of the available, active tests, afterward enabling them to start solving it. After the student is done solving the test, the results are shown. Of course, only multiple-answers questions results will be shown, whereas the essay questions still need to be reviewed by a professor which created the test. If the chosen answer is correct, it will be shown in green. If the chosen answer is not correct, it will be shown in red, while the correct answer will be shown in blue.

## 4. ROI ONTOLOGY

In computer science, an ontology encompasses the display, formal naming and defining categories, properties and relations between the concepts, data and entities that corroborates one, many or all domains of the discourse. Simply put, an

ontology is a way of representing properties of a domain and their relations by defining a set of concepts and categories that represent the subject.

Regions of interest would in this case be parts of the screen that have useful information for the semantic of the application. Defining these elements with an ontology would give an additional ability to understand and use their semantics.

The root class of the ROI ontology is named Roi. It consists out of five classes: Name, Start, End, Question, Time. Class Name uniquely defines ROIs. Start and End classes are start and end points of an ROI, that is, the top left and the top right corner of the rectangle respectively. Class Question references the question that ROI belongs to, while the Time class represents fixation time.

This ontology gives a valuable insight into how the spatial layout of a question can impact students' understanding of it. For example, if the student has to revert their gaze multiple times to someplace that is distanced from the firsthand ROI, it could impact their focus, thus reducing their ability to understand the question correctly.

Once the ROI is presented by the ontology, all of this information can be accessed and possible adjustments can be made for future iterations.

## 5. IMPLEMENTATION

The application itself is separated into three parts. The presentation layer, that is, the frontend of the application has been developed in the Angular framework. This framework enabled implementing the single-page pattern, which proved to be useful for this platform, especially in developing test-creation functionality.

The business logic of the platform, that is, its backend was implemented in SpringBoot. The data itself was stored in a MySQL database. SpringBoots security came in handy to protect from unauthorized access to the tests.

Third, and last, part of the application is the proxy module. Its role was to connect the application with the eye tracker camera while getting the data from the frontend. The data is parsed in this module and sent to the backend after the test is finished.

For the eye tracker camera, we used a Gazepoint Eye Tracker.

## 6. CONCLUSION

The problem that was being solved in this paper concerned the implementation of an online platform on which examinees could be tested while their eye movement was being tracked. By investigating what measures are important in determining how an examinee understood the exam, we concluded that gaze fixation and saccade amplitudes, their lengths and number play can give valuable information.

For example, an examinee spends an excessive amount of time fixating on one question is a good indicator that there is something wrong with that question. Maybe the question is poorly set together, or maybe there is not enough clarification. The spatial layout of the exam is also very important. That problem can be seen by looking at the fixation and saccades combined. If there an examinee keeps moving their gaze from one point to another more than usual, that could be an indicator that the spatial layout of the exam is not correct and should be readjusted in the next iteration.

ROIs play a big role in deciding whether an exam was correctly put or not. By modeling them with an OWL ontology, we were able to register all the ROIs, their place on the screen and the time spent fixating on it for every individual student.

The platform itself allows professors to create tests and see their result in a heatmap, while the students can access active tests and solve them.

## 7. REFERENCES

[1] Andrzejewska M., Skawińska A. (2020): Examining Students' Intrinsic Cognitive Load During Program Comprehension – An Eye Tracking Approach. In Bittencourt I., Cukurova M., Muldner K., Luckin R., Millán E. (eds) Artificial Intelligence in Education. AIED 2020. Lecture Notes in Computer Science, vol 12164. Springer, Cham. https://doi.org/10.1007/978-3-030-52240-7_5

[2] Wang X., Lin L., Han M., Spector J. M. (2020): Impacts of cues on learning: Using eye-tracking technologies to examine the functions and designs of added cues in short instructional videos. Computers in Human Behavior 107, 106279, 0747-5632. https://doi.org/10.1016/j.chb.2020.106279

[3] Ujbanyi T, Katona J., Sziladi G. and Kovari A. (2016): Eye-tracking analysis of computer networks exam question besides different skilled groups, 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 000277-000282, doi: 10.1109/CogInfoCom.2016.7804561.